

FREQUENCY DOMAIN SIGNAL PROCESSING FOR ACOUSTIC SENSORS

1. REPORT DOCUMENTATION PAGE (Cover Page)

2. PROJECT OBJECTIVES

The general objective of this contract was to demonstrate the feasibility of separating cruise missile class targets when there are multiple targets observed by the acoustic sensor. Several programs (e.g., Steel Rattler, FASDR) have addressed the use of small, remotely deployed acoustic sensors for collecting early warning and rough track on cruise missiles. Typically, the quality of track expected is not precise enough for use as fire control data, but rather is useful to cue fire control sensors. Acoustic sensors in this role significantly increase the time available to engage targets and provide very early warning of impending attack. Field experimentation has indicated that using acoustic sensor data to develop tracks of single targets is feasible and practical [Ref: Steadman, R., "Air Deliverable Acoustic Sensor (ADAS), Multi-Node Data Fusion for Tracking Fast Aerial Targets", Textron Systems Division Briefing, October 1996]. The problem that had not been adequately addressed until this contract effort is how to develop tracks of multiple targets (cruise missiles) from acoustic data.

The specific objective of this Phase I SBIR was to develop the signal processing techniques that will enable adequate track data to be developed from acoustic observations of multiple cruise missile targets. The specific objectives of the contract were:

- Acquire acoustic data on multiple targets that operate in the cruise missile flight regime.
- Develop and implement processing algorithms for parsing the time-domain data into segments for transformation into frequency domain data.
- Test several alternative signal processing techniques against the frequency domain data to identify the best candidate for discriminating distinctly different sources via their frequency change patterns, to identify data components that can be used to form tracks of each of the multiple targets.
- Conduct an automated demonstration of the separation of acoustic data from multiple targets into several distinct tracks with quality sufficient to enable hand-over to another surveillance sensor or to a fire control sensor.

3. WORK PERFORMED

Task 1 – Acquire acoustic data on multiple targets.

Acoustic data was secured from the National Ground Intelligence Center (NGIC). The data was provided on Hi-8 mm videocassette footage shot with a camcorder at an air show.

The footage was reviewed and three vignettes of interest (one or two jet aircraft flying past the camera at an altitude and speed commensurate with cruise missile flight profiles) were identified.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</p>			
1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE 15 February 2000	3. REPORT TYPE AND DATES COVERED Final Report 16 Apr 99 - 15 Feb 2000	
4. TITLE AND SUBTITLE Frequency Domain Signal Processing for Acoustic Sensors		5. FUNDING NUMBERS Contract No: DASG60-99-M-0091	
6. AUTHOR(S) John A. Dennis Ilya Schiller Tim A. Patterson		Purchase Req No: DR 9A92/400-01	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Radiance Technologies, Inc. 500 Wynn Drive, Suite 504 Huntsville, AL 35816		8. PERFORMING ORGANIZATION REPORT NUMBER SBIR99.1-0002	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) SMDC-TC-AI P.O. Box 1500 Huntsville, AL 35807-3801		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION/AVAILABILITY STATEMENT (see Section 5.3b of this solicitation) Approved for public release; distribution unlimited		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Report developed under SBIR contract for topic BMDO99-003A. This contract effort demonstrated the feasibility of separating cruise missile class targets when there are multiple targets observed by a ground positioned acoustic sensor. Several programs (e.g., Steel Rattler, FASDR) have addressed the use of small, remotely deployed acoustic sensors for collecting early warning and rough track on cruise missiles. Typically, the quality of track expected is not precise enough for use as fire control data, but rather is useful to cue fire control sensors. Acoustic sensors in this role maximize the time available to engage targets and provide very early warning of impending attack. Field experimentation has indicated that using acoustic sensor data to develop tracks of single targets is feasible and practical. One of several challenges that have not been adequately addressed until this contract effort is how to develop tracks of multiple targets from acoustic data. This contract effort demonstrated algorithms that will identify signals of multiple targets and will parse individual target data from acoustic signals generated by multiple targets.			
14. SUBJECT TERMS Acoustic Signal Processing, Separating Multiple Targets, Cruise Missiles		15. NUMBER OF PAGES 12	
		16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

Standard form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

REF C

1. Footage with a single aircraft flying past the camera.
2. Footage with two aircraft flying past the camera in formation.
3. Footage with two aircraft where one aircraft continues to fly level and the other breaks and starts a near vertical climb.

To retrieve this data, a video camera was connected to a PC/Windows 98 platform computer by a 1/8 " stereo patch cord via the computer's sound card. Each vignette identified above was played back on a camcorder patched through the sound card on the computer (Figure 1)

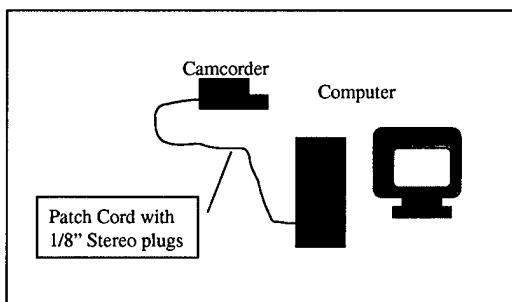


Figure 1. Setup for Recording the Vignettes

A shareware application called Cool Edit was used to record and digitize the data and save it to ASCII text files for input into other processing programs. Cool Edit is a digital audio recording and editing software application. Each vignette was recorded and digitized with Cool Edit from the camcorder playback. In Cool Edit, we used 16-bit stereo digitization with 16000 samples per second. The sound portion of each vignette was saved on the computer's hard drive in ASCII text

format as well as a Windows *.wav sound file format. The Cool Edit screen shot in Figure 2 shows the amplitude plot of each channel of the 3.524 second vignette of the single aircraft.

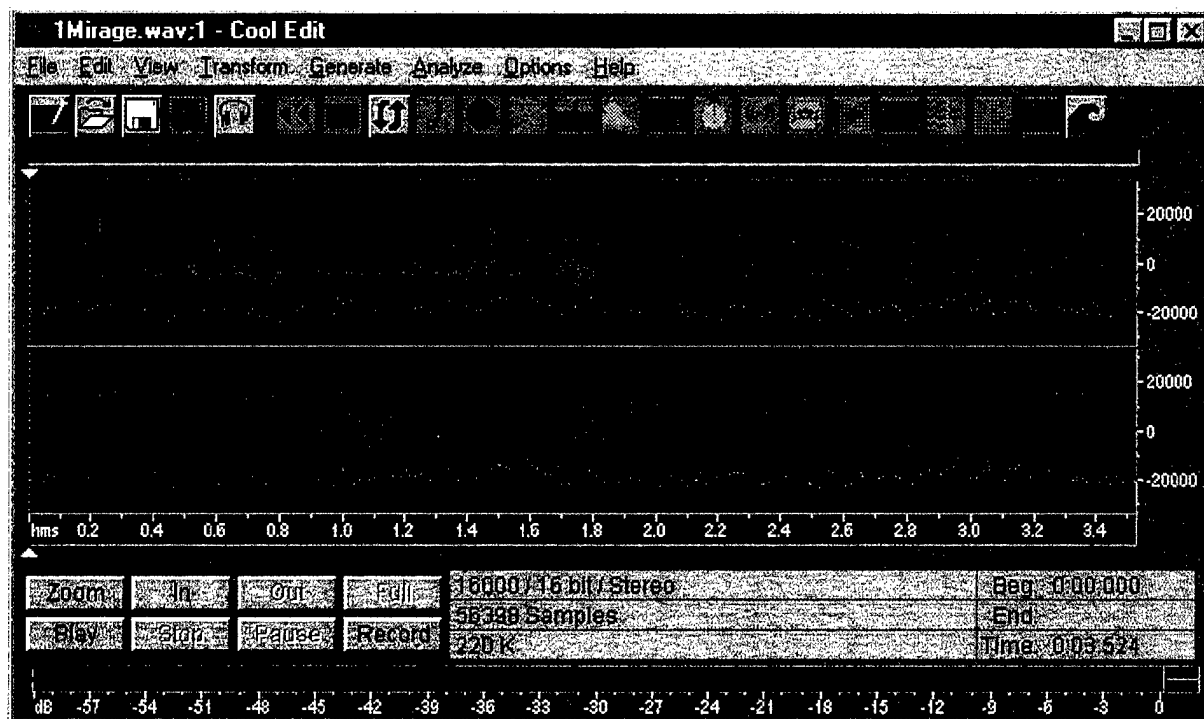


Figure 2. Example Cool Edit Processing Window

We also used Cool Edit to make preliminary Fourier Transforms of these data to reveal the frequency content as a function of time. Figure 3 is a Cool Edit FFT (Fast Fourier Transform)

screen. We used the Cool Edit FFT outputs to determine where in the frequency spectrum the acoustic energy resided.

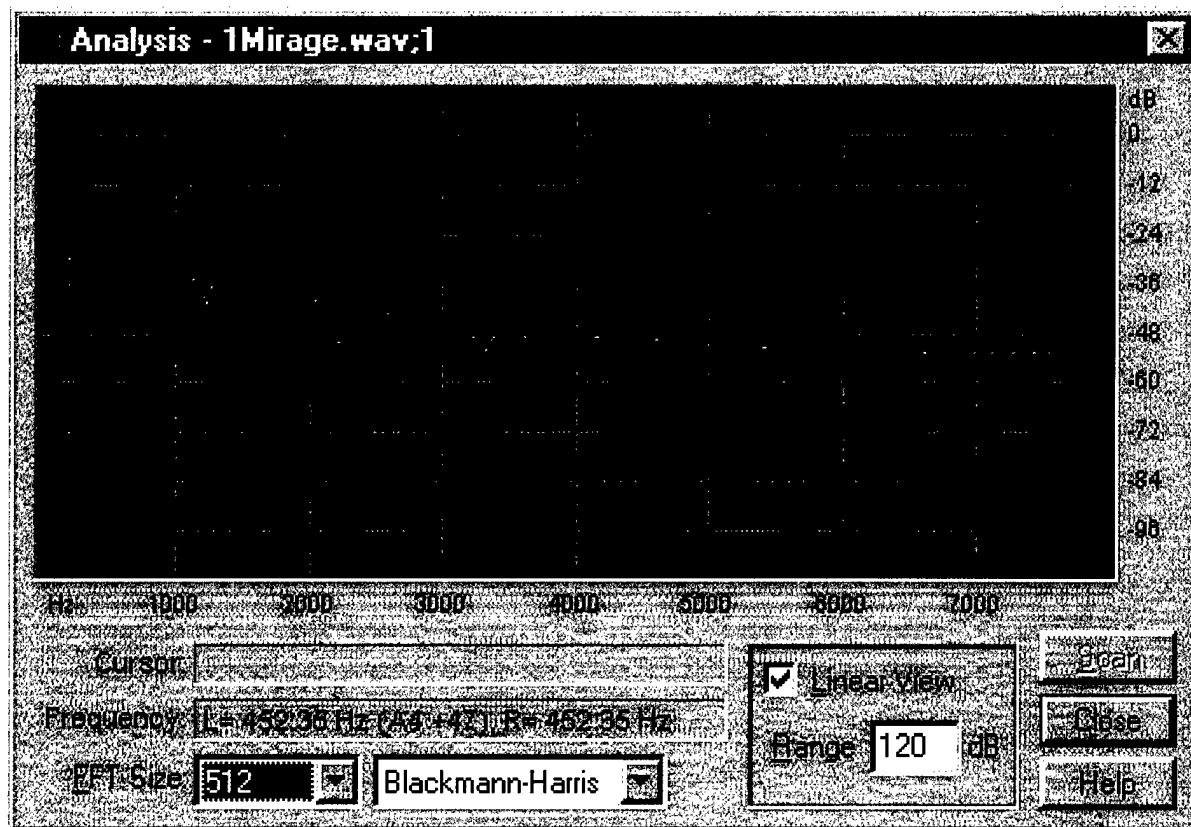


Figure 3. Cool Edit FFT Screen Shot

The data that was originally digitized using Cool Edit was then written to a data file in ASCII format, so that it could be read by processing routines developed in IDL (Interactive Data Language). IDL is a comprehensive computing environment created by Research Systems, Inc. for the interactive analysis and visualization of data. It integrates a powerful array based computer language, numerous mathematical algorithms and graphics utilities. This language allows for rapid code development, interactive data visualization, and provides several built-in functions for graphical user interface (GUI) development. We wrote a simple processing algorithm in IDL to apply the Fourier transform to the data and the time frequency history. We applied this IDL routine to the ASCII files generated from the data reduction effort. A computer program was created in IDL to take the FFT of a single channel of the digitized sound data to determine regions in the spectrum that may be of interest. Figure 4 shows the FFT of the entire data sample for all three target (aircraft) vignettes. These plots show that the bulk of the acoustical energy is below 2000 Hz with a peak at around 400 Hz for each vignette. This fact was used in subsequent processing steps to help refine and “zero-in” on the signals of interest.

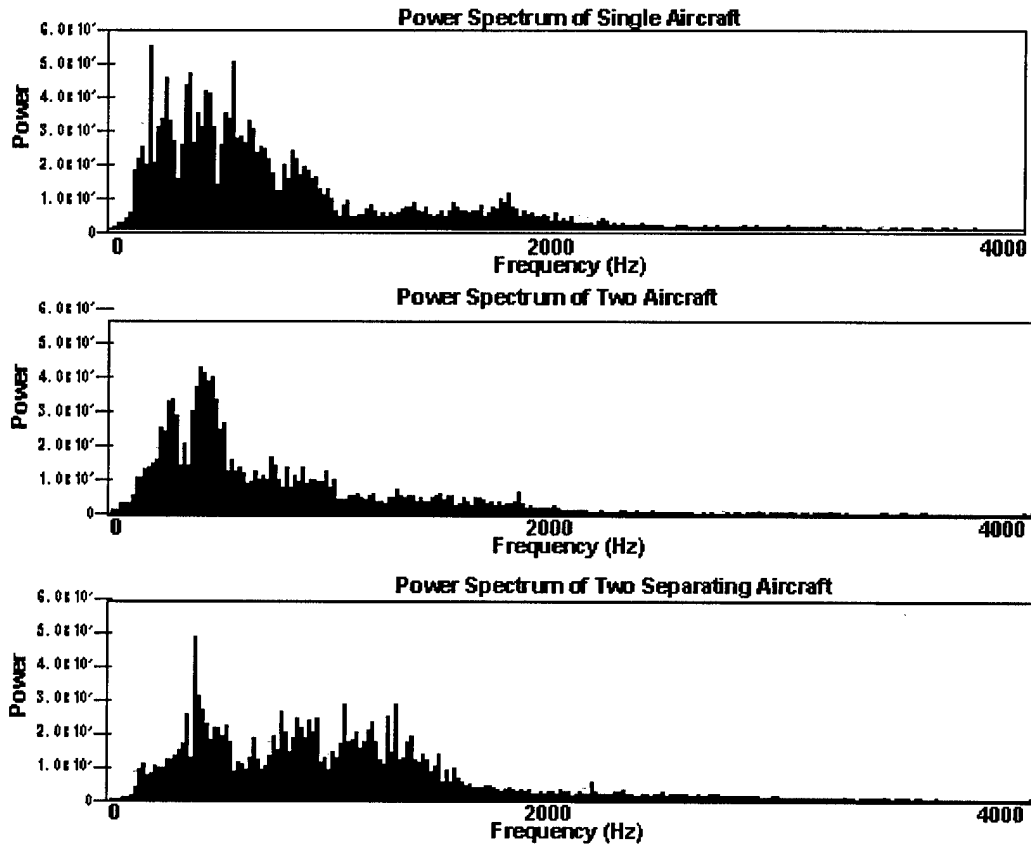


Figure 4. FFT of Entire Data Set for Each Vignette

Task 2 – Develop and implement parsing algorithms. The data was then “pre-processed” into the frequency domain with IDL using several techniques to determine which technique provides the best reasonable approximation of what could be done rapidly in the field and still highlight the frequency separations of two targets. The technique that was employed was a windowed Fourier Transform. This technique allows for a time history of the FFT over the entire vignette to be constructed thus providing a spectro-temporal graph. Specifically, this technique was implemented by loading the ASCII data of the vignette into an array and processing only the right channel. A FFT was taken by starting with the first sample and the following 1023 contiguous time samples. The time sample interval for all vignettes was 62.5 microseconds. Given that there was 1024 samples, this provided a power spectrum with a frequency resolution, Δf of

$$\Delta f = \frac{1}{1024 * 62.5 * 10^{-6}} = 15.625 \text{ Hz}$$

To produce the next FFT, 128 time samples were skipped from the beginning time step of the previous FFT. This gives a time step of

$$\Delta t = 128 * 62.5 * 10^{-6} = 0.008 \text{ sec.}$$

Then the next contiguous 1023 time samples were taken from this starting point and used with an FFT to produce the power spectrum for this time step. This process was continued until the last full 1024 sample data set for that vignette was used.

The pictures below are spectro-temporal profiles produced by the processing technique outlined above. Figures 5, 6, and 7 are spectro-temporal plots for the single target vignette, the two targets vignette and the vignette with two targets separating. In the single target case, Figure 5, there is a spectral peak (the darkest pixels, outlined by a green dashed oval) at around 200 Hz. The frequency of this peak increases to about 400 Hz at 2 seconds and then falls to 100 Hz.

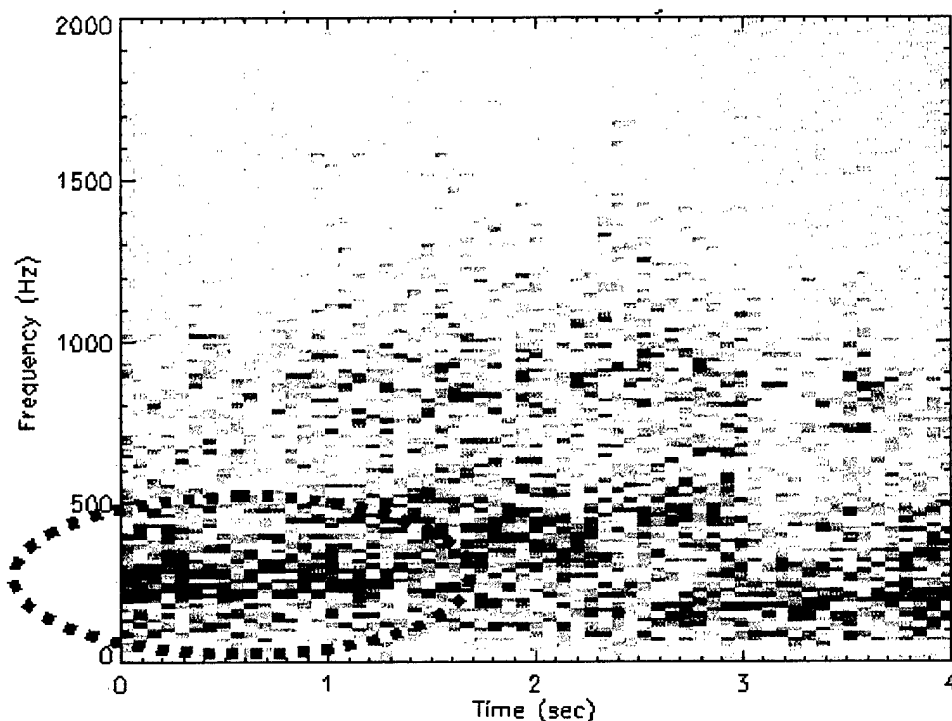


Figure 5. Spectro-Temporal Plot for Single Target Vignette

This trend indicates the Doppler shift as the target passes the sensor (in this case the camcorder). In the two-target vignette, Figure 6, there are two spectral peaks, one that starts at 500 Hz and the other at 200 Hz. In Figure 6a, these spectral peaks are highlighted with a green dotted oval and a blue dashed oval. At approximately 1.5 seconds, some higher frequency signals appear (red dotted oval in Figure 6a). These signals correspond to human voices that are present on the original tape. At approximately 4.5 seconds there is a shift in frequency of the two peaks, corresponding to the target passing the sensor.

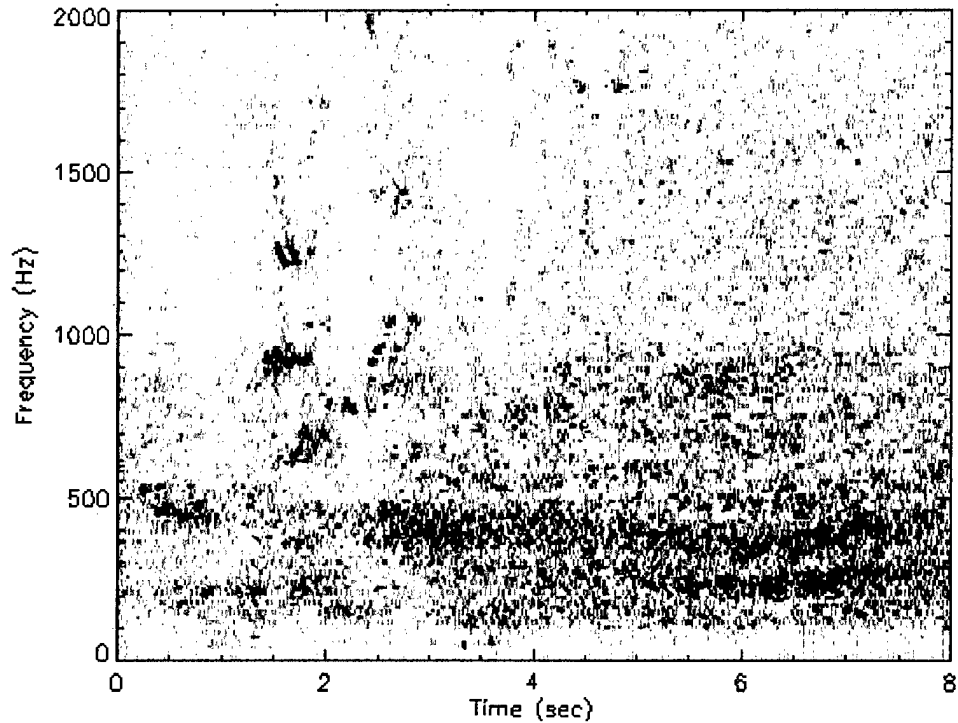


Figure 6. Spectro-Temporal Plot for Two Target Vignette

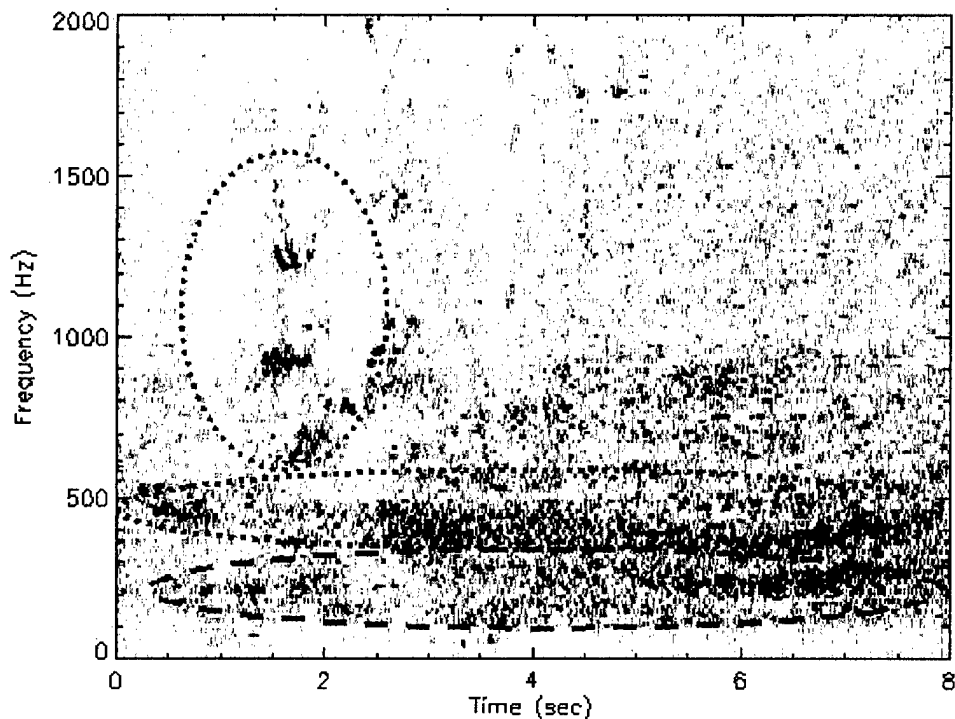


Figure 6a. Spectro-Temporal Plot for Two Target Vignette, Annotated

In the vignette with two targets separating (Figure 7), there are three spectral peaks. The one at around 200 Hz represents the one target that continued flying level towards the sensor. This is similar to the data in figure 5. The higher frequency peaks correspond to a closer target that broke away from the other at about 6.5 seconds.

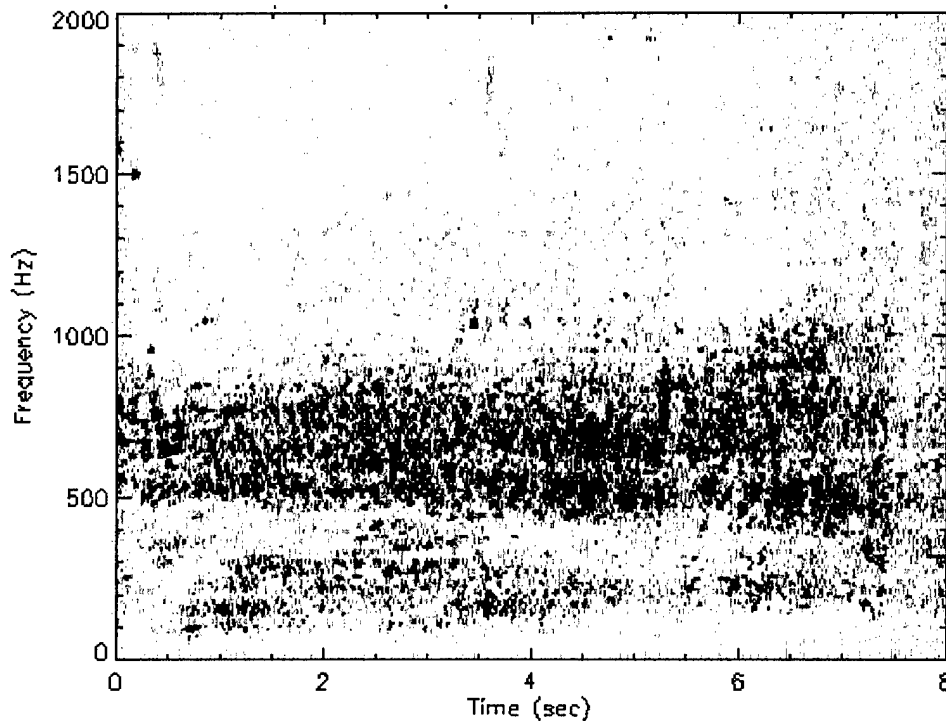


Figure 7. Spectro-Temporal Plot for Vignette with Two Targets Separating

In Figures 5 through 7, the vertical axis is frequency, the horizontal axis is time, and the color (degree of blackness) represents power of the signal within the frequency bins (darker represents more power).

To prepare these data for use in the software, the collection of FFTs were coded according to their magnitude in order to extract the feature of interest (magnitude at each frequency over time). Each time slice then had a collection of magnitude bins coded to represent the magnitude at that frequency. Each coded time slice is a vector that has both time and frequency features. These features are used in the separation software to cluster the data into classes.

Task 3 – Identify the best separation techniques. We selected Vector Quantization as the most appropriate technique for separating the target signals in software.

Vector Quantization (VQ) is a system for mapping or quantizing a sequence of vectors into representative classes. This technique capitalizes on any underlying structure of the data being quantized. VQ encompasses many statistical methods of quantizing. Recently adaptive techniques have been used in quantization of data for VQ, thus incorporating unsupervised learning principles. Unsupervised learning means that the representative data are clustered in classes according to “closeness of features”. Once the “closeness of features” is defined, the learning algorithm clusters or classifies the data automatically.

VQ based methods are particularly suited for speech and acoustic signal processing. Measured multidimensional and time-dependent acoustic data samples are viewed as n -dimensional vectors. A cluster or class in that vector space corresponds to a group of vectors

with similar "closeness of features" or measurement characteristics. The measurement characteristics in this feature space are a basis for identification of classes.

The set of vectors that are mapped to a particular class is referred to as the cluster of the class. The cluster is a group of samples (vectors) with a close relative distance to each other in terms of a metric.

The purpose of a metric for "closeness of features" is to separate important features in samples. The metric is used to measure the distance or distortion between data samples (vectors). Several metrics may be considered depending on the problem at hand. The use of the metric will affect the classification since each metric emphasizes some features more than others. Experience has shown that the Euclidean distance is a practical measure for feature separation:

$$d(X,Y) = \sqrt{\sum (x_i - y_i)^2}$$

A vector that is representative of the cluster is sometimes called the centroid. The collection of all centroids represents all possible classes. The goal of the VQ algorithm is to separate data into representative classes, find the centroid of each class and cluster the data around them so the total distortion is minimized.

The VQ clustering algorithm consists of two stages. The encoding or classification stage is when a new input vector is placed in a cluster. The encoding computes the metric, comparing the new input vector to each existing centroid. The input vector is assigned to a cluster with closest distance to that cluster's centroid. The encoding stage is rapid and can be accomplished in near real-time.

The second stage, called training, determines where the centroids of the clusters are. This is much more complicated task. The centroids should be defined in such a way that the total distortion of the data is minimized. Both statistical (Gray, 1984) and neural network (Ahalt, 1990) techniques may be used. The total distortion of the data is computed as the sum of all individual vector distortions compared to their corresponding centroids.

$$DM = \sum_{i=1}^T d_i$$

where T is the total number of vectors in all the clusters.

Clustering the data by similarity (measured via a metric) enables a variable scale of resolution (Romesburg, 1984). Similarity may be so broad that all samples are put into one cluster, or so narrow that every sample defines a unique cluster. Resolution is a user-defined property, defined in most instances by use of a training set in which the desired outcome is known in advance.

When the number of clusters is known in advance, a user-defined definition of resolution is unnecessary, and the process consists of finding the specified number of clusters and assigning data to the appropriate cluster

The tree structured VQ method (Gray, 1984) allows recursive construction of the centroids. In this method the algorithm starts with a few centroids and performs what is called "splitting". Each cluster's centroid is "split" into two adjacent centroids. Each vector's distortions in that cluster are calculated to each of the two centroids and then each vector is assigned to one

centroid or the other based on the minimum distance. Then the training is performed. As vectors (data) are added, the centroids start to drift. Checking the distortion between centroids may recombine clusters. When the distortion between two centroids falls below a predetermined level, the two clusters are combined and a new centroid for the combined centroid is computed. This process is commonly called "pruning".

The process for finding the set of centroids so that total distortion is minimized is called training. This automated process is summarized in Figure 8. The final step of this process, labeled Step 6 or Step 6a, is different when the number of clusters is specified from when the number of clusters is unknown. When the number of clusters is not known, the maximum tolerable distortion is the user-defined parameter that affects the number of clusters. Often, inspection of a preliminary result will lead to a refinement of the desired distortion level for another iteration.

Step 1	Introduce several centroids and place them randomly in vector space
Step 2	Introduce a data vector and compute its distortion to each centroid.
Step 3	Assign the data vector to the cluster represented by the centroid with minimum distortion.
Step 4	Modify the centroid to accommodate the added data vector.
Step 5	Conduct Splitting and Pruning to ensure the proper number of clusters.
Step 6	Repeat steps 2--5 until all data is used in clusters
Step 6a	Repeat steps 2--5 until all data is used and the total distortion is less than the desired value.

Figure 8. Summary of VQ Process.

To prepare the data described in Figures 5-7 for use in the classification software, we coded each FFT according to magnitude. Each time slice then had a collection of magnitude bins that are coded to represent the magnitude at each frequency. Each time slice is a vector that has both time and frequency features for use in the classification software.

Given that the classification software could identify a multi-target case, the next step is separation of the data into separate files of single target data. To accomplish this, we exploited the original spectro-temporal data with the algorithm shown in Figure 9.

Step1	Select a small time slice.
Step 2	Examine the frequency spectrum to see if double-peakedness exists
Step 3	If not, widen the time slice
Step 4	When a time slice is found with two peaks, separate the data into two bins, one for each target
Step 5	Concatenate each bin to the two bins found in the previous time slice (if any)
Step 6	Continue until end of data
Step 7	Upon end of data, there will be two bins of time-frequency data, one for each target

Figure 9. Algorithm for Separating Multi-target Data

For the data in Figure 6, an example of the time slices is shown in Figure 10. A wider slice is required between 0 and 1 second in order to capture two peaks of signal strength, whereas

at time 3.5 and greater, smaller time intervals will capture two peaks. The separation occurs at about 350 Hz shown as dashed crossbars in each example time slice.

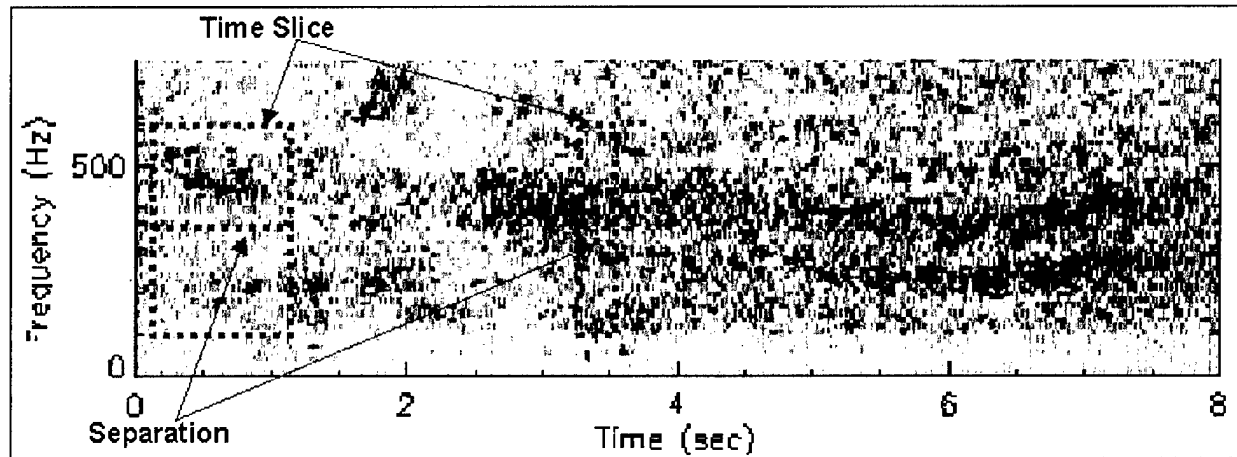


Figure 10. Example Time Slices in Multi-Target Data

Once the data is processed and concatenated, the processor has separate files for each target (Figure 11), ready for combining with data from another sensor to build a track file.

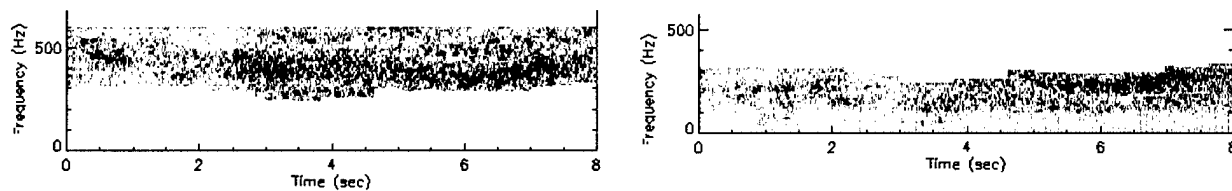


Figure 11 Separation of Multiple Target Signatures

Task 4 – Conduct an automated demonstration.

A training set for the neural net classifier software (VQ) was devised by selecting every even numbered time sample (every other .008 second) from the data sets for one-target (Figure 5) and two targets (Figure 6). The odd numbered time samples (every other .008 second) from these data sets were used as the first test set.

We knew in advance that two clusters were appropriate, one cluster for a one-target class and one cluster for a two-target class. The same process, with total distortion as the controlling feature, could be used to identify how many targets are present in a multiple target environment.

During training (even numbered samples from the one-target and the two-target data) we started with the centroids for five clusters and determined which cluster correspond to the one-target class and which to the two-target class. We used the Euclidian distance described above as the similarity measure.

During testing (odd numbered samples from the one-target and the two-target data) each vector was compared to its nearest clusters in 64-D space to assign the closest one based on the Euclidian distance similarity measure.

Since there is risk that our test set is correlated with our training set, we tested our algorithm on a third test set (the data described in Figure 7, two targets separating) was pre-processed into the frequency domain for use in the neural network classifier. This data was applied to the algorithm previously trained with the even numbered samples from the one-target and two-target data, and previously tested with the odd numbered samples from the same data.

When the algorithm identified a multiple target case, the power peaks in the spectro-temporal data were used to identify the separation (in frequency) between the signature of one target versus the other. The gap between the two peaks was identified as the separator, with the principal signature of one target occurring above the gap, and the other below the gap. A tracking algorithm then could operate on the two subsets of data independently.

4. RESULTS OBTAINED

The trained network achieved almost 91% success in correct classification of the one-target data, and almost 94% success in correct classification of two-target data. In Figure 12, the numeric values in the colored (yellow for one target and blue for multiple targets) represent the number of times that the algorithm correctly identified the correct number of targets in the data.

Acoustic Data from One Target			
Number of Times Algorithm Identified Source as:			
Single Target	Multiple Targets	Single Target	
58	6	4	

Figure 12. Results of Initial Test

Each classification result, or an instance of classification, represents a very short interval of time. If an acoustic array were monitoring the flight of a target, the total number of classification results may be combined to provide an overall estimate of the number of targets. Neither test result is ambiguous (58-6 and 64-4), therefore either result would support an alert from the acoustic sensor system identifying a single target or multiple targets. In addition to being unambiguous, the reports would be correct.

Since there is significant risk that the initial test set is correlated with the training set, a second test set (the third section of original data, containing two targets, and represented in Figure 7) was pre-processed into the frequency domain for use in the neural network classifier.

Figure 13 is the result for the test set that has two targets that separate in different directions. All parameters are the same as in the previous test. In every classification instance, the algorithm correctly classified the data as representing multiple targets.

		Single Target
		0

Figure 13. Results of Confirmation Testing

This confirmation test result is even less ambiguous than was the initial test, and is also correct. Based on the testing done in this effort, an acoustic sensor array using these techniques could reliably report that multiple targets had been observed. The separation of targets in this

data set was computed to occur at about 650 Hz, and can be easily identified visually in Figure 13, which is the data set from Figure 7, with the separation indicated with red dotted lines.

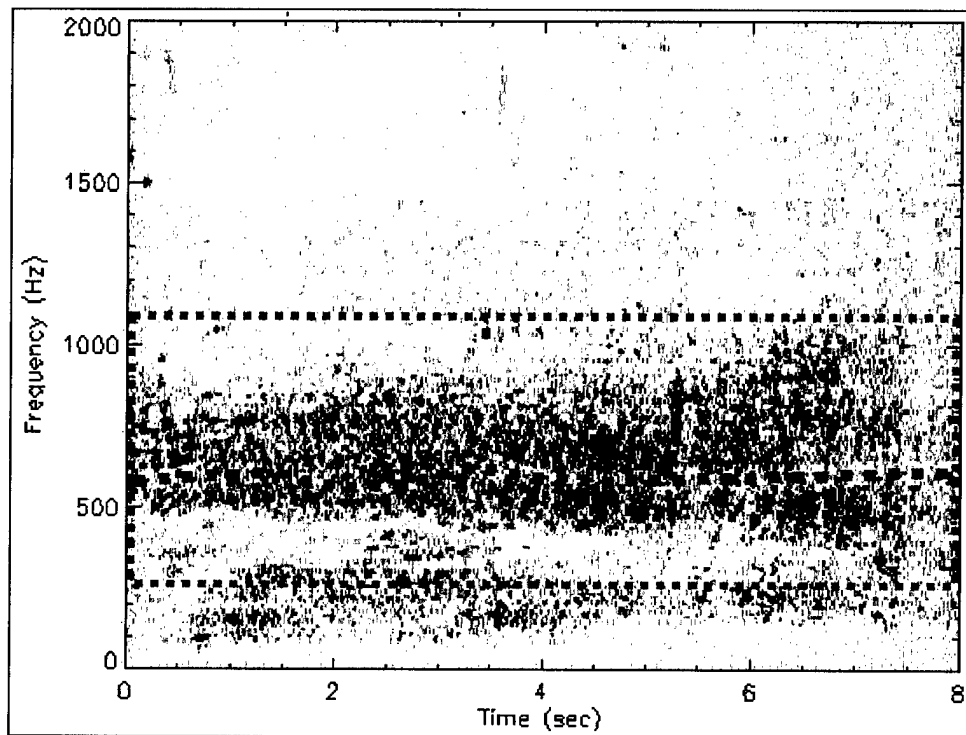


Figure 13. Separation of Targets for Two Targets Separating

5. ESTIMATE OF TECHNICAL FEASIBILITY

The software used to conduct this demonstration can easily be configured onto a digital signal processing board and added to any processor used to process acoustic signals to define tracks from targets such as cruise missiles. A capability to recognize (greater than 90% of the time, as demonstrated in this effort) that there are multiple targets rather than just one can provide a substantial improvement in military capability.

6. REFERENCES

1. Ahalt, C. Stanley, Krishnamurthy, K. Ashok, Chen, Prakoon, Melton E. Douglas, Competitive Learning Algorithms for Vector Quantization, *Neural Networks*, Vol. 3, pp 277-290, 1990
2. DeSieno, D., Adding a Conscience to Competitive Learning *IEEE International conference on Neural Networks*, pp 1117-1124. 1988.
3. Gray, R. M., Vector Quantization, *IEEE ASSP Magazine*, 1(2), pp 4-29 1984.
4. Grossberg, S. Competitive Learning: From Interactive Activation to Adaptive Resonance. *Cognitive Sciences*, 11, pp 23-63. 1987
5. Kohonen, T., Self-Organization and Associative Memory, Springer-Verlag, 1984
6. Kohonen, T., "Improved versions of learning vector quantization", In Proceedings of the International Joint Conference on Neural Networks, pages I 545-550, San Diego, June, 1990a.

7. Kohonen, T., "The self-organizing map", *Proceedings of the IEEE*, 78(9):1464-1480, 1990b.
8. Kohonen, T., "Statistical pattern recognition revisited", In *Advanced Neural Computers*, pages 137-144, 1990c.
9. Kohonen, T., Kangas, J., Laaksonen, J., Torkkola, K., "LVQ_PAK: A program package for the correct application of Learning Vector Quantization algorithms", *Proceedings of the International Joint Conference on Neural Networks*, pages I 725-730, Baltimore, June 1992. IEEE.
10. Kohonen, T., *Self-Organizing Maps*. Springer-Verlag, Heidelberg, 1995.
11. Krishnamurthy, K. Ashok, Ahalt, C. Stanley, Melton E. Douglas, Chen, Prakoon, "Competitive Neural Networks for Vector Quantization of Speech and Images", *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 8, pp 1449-1457, Oct 1990
12. Makhoul, J., Roucos, S., Gish, H., "Vector Quantization in Speech Coding", *Proceedings of IEEE*, Nov. 1985.
13. Nasrabadi, N. M., King, R., A., "Image Coding Using Vector Quantization: A Review", *IEEE on Communications*, Aug. 1988.
14. Romesburg, H. Charles, *Cluster Analysis for Researchers*, Lifetime Learning Publication, 1984
15. Steadman, R., *Air Deliverable Acoustic Sensor (ADAS), Multi-Node Data Fusion for Tracking Fast Aerial Targets*, Textron Systems Division, October 1996.
16. Wan, E. A., Ning, P., Widrow B., *Neural Tree Structured Vector Quantization*, *International Joint conference on Neural Networks, IJCNN-90-Wash-DC*, v II, pp 267-270. January 1990.